

Maemo 5 Fremantle alpha SDK

Kate Alhola 7.4.2009



Introduction



Kate Alhola

- Maemo Chief Engineer at Forum Nokia
- Participating Qt maemo port project, several maemo projects in garage.maemo.org
- Long term Open Source developer, first contributions 8-bit microprocessor in early 80's
- Linux kernel driver from early 1.x kernels
- Katix RTOS with IP stack for PC, 68K and PPC
- Multiple GUI applications with Qt and GTK (and X11/Athena, Motif ...)
- Before Nokia, long career embedded Linux and RTOS related development in small subcontractor companies
- Numerous embedded HW designs

Forum Nokia

- Nokia 3rd party developers organization
- Consultancy, marketing, business development, university support, technical support, developer events, prototype loaning, etc

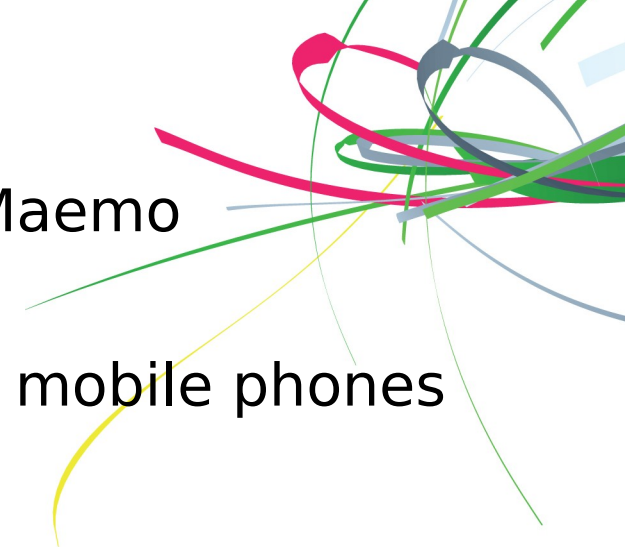
Maemo Linux for mobile devices

- For Pocket/handheld sized mobile devices
- Internet oriented devices, best browsing experience that you can get in pocket size device
- Small form factor 800x480 screen
- Finger optimized touch screen
- Takes advantage of device's multimedia capabilities including DSP all device connectivity features
- Strongly optimized power management allows long battery life and small device battery size
- On screen virtual keyboard or mobile optimized small size qwerty keyboard



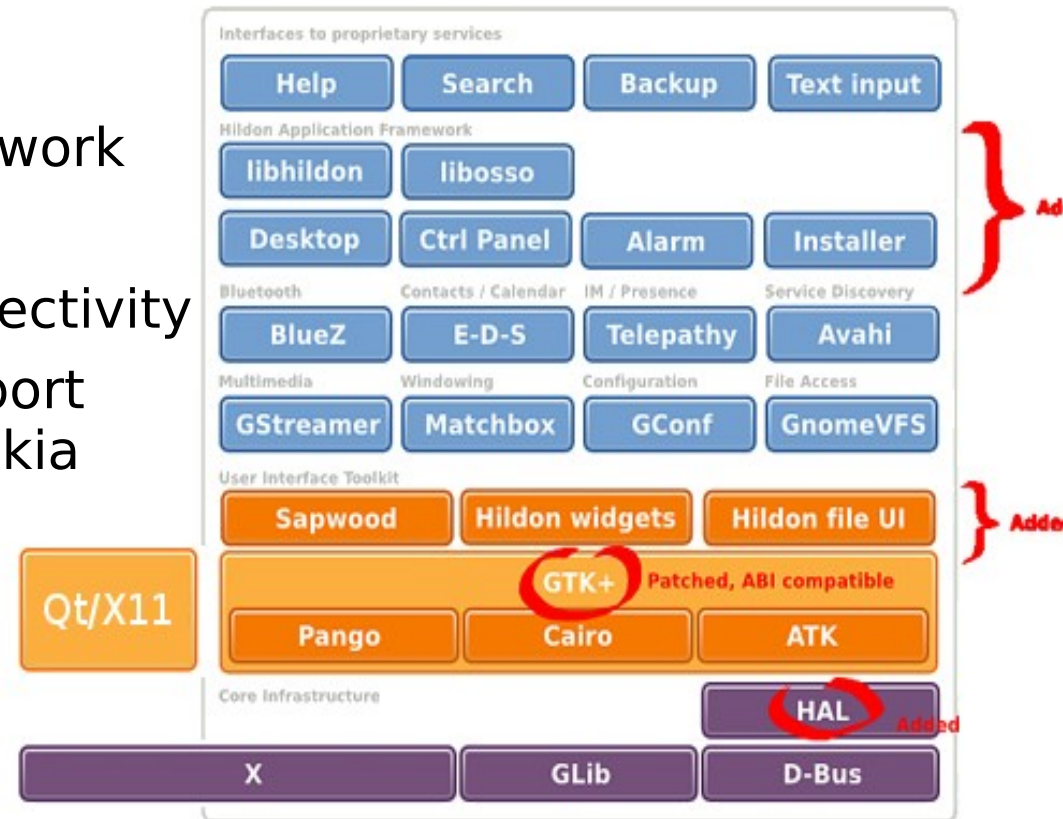
What maemo is and is not

- is optimized for mobile pocket size devicesMaemo
- Maemo is not for desktops
- Maemo is not for small low resolution screen mobile phones
- No stuff that wont fit in your pocket
 - No need for hard disk
 - No need full size keyboard
 - No mouse
 - No need for big heavy battery
 - No need for big screen
- No stuff that spoils your internet experience
 - No keyboard only navigation
 - No mini siz e low resolution screen
 - Not limited to one toolkit



Maemo 4 (Diablo)

- Current Maemo production release
- Nokia devices supported: N800 & N810 & N810 WiMAX edition
- OMAP 2 processor
- GTK+ toolkit with Hildon UI framework
- Maemo desktop UI
- WLAN, Bluetooth and WiMAX connectivity
- Qt 4.4 with Hildon framework support as community port from Forum Nokia



Maemo roadmap

UI Framework progression



Fremantle
Hildon & GTK+ evolved



Harmattan
Qt integrated

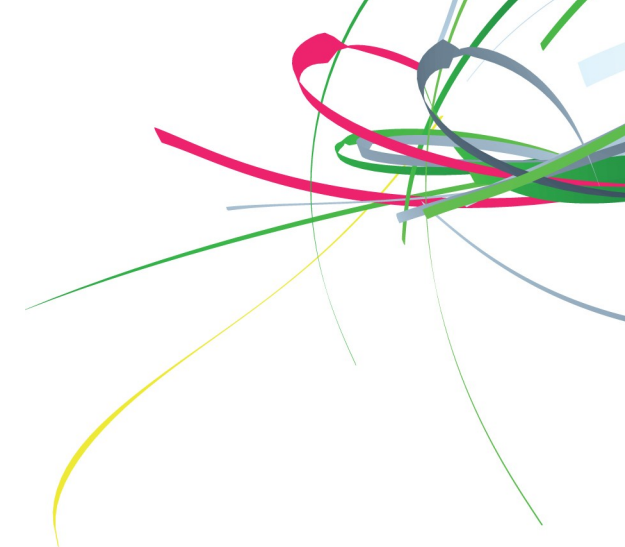


Consolidation of essential parts of the platform

Image: *Industrial-Glade*, by John Brian Silverio. CC Attribution-Non-Commercial-No-Derivative-Works License

Maemo 5 Fremantle

- New generation of Maemo
- New UI style
- Accelerated graphics with OpenGL-ES2
- Animated Ui technologies
 - Clutter
 - Qt4.5 from Forum Nokia
 - Compositing window manager
- Some new API's
- High resolution camera
- Omap3 architecture
- HSPA (High Speed Packet Access / Cellular 3G connectivity)

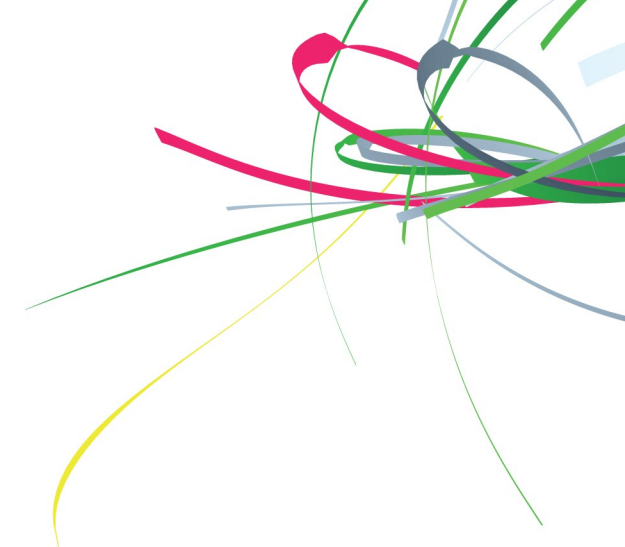


Improved usability

- Designed for finger usage, not depending of stylus
- No small UI elements
- Scrolling from content pane, not small scrollbars
- Improved dialog layout
- Compositing window manager

Where we are now

- Maemo 5 Fremantle
- First pre-alpha SDK released in Nov 2008
- Alpha SDK released March 2. 2009
- Beta SDK coming soon



Fremantle alpha SDK means to developer ?

- **alpha is first release that can be used as software development**
- OpenGL-ES 2.0 libraries allow to write accelerated animated graphics
 - Qt graphicsview and GL-Widget
 - Clutter
- New hildon desktop with composite window manager
- Beagleboard allows to run maemo in real Omap3 prosessor

Top framework level changes from Diablo

An abstract graphic in the top right corner consisting of several overlapping, curved lines in green, yellow, and red, resembling a stylized network or a dynamic design element.

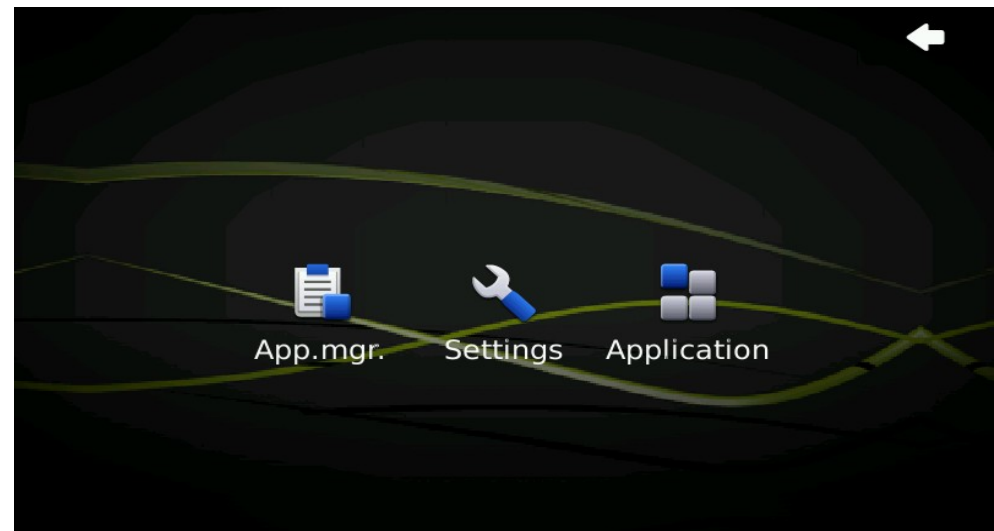
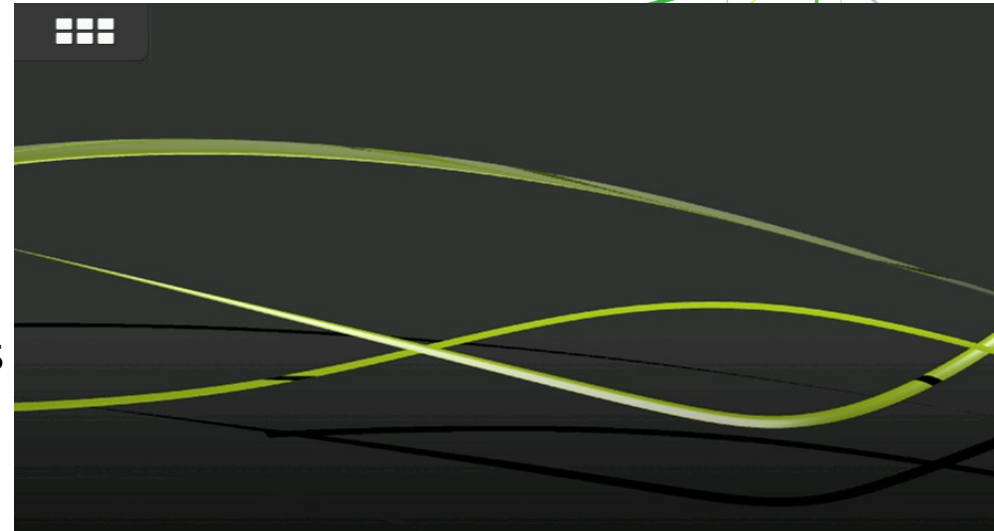
- Removal of left side Task Navigator and plugins.
- Removal of stylus keyboard
- New design for task switching and task handling
- Renewal of Home and Status bar
- New design for incoming event previews and indications
- Widgets and application space renewal
- UI style changes.Eg: Navigation logic in applications, visual style of dialogs, menus etc

NEW API's

- Location API: methods to build location-aware applications.
- City Information: methods to obtain information about cities, including city name, country name, and country code.
- Time management: an interface for handling time change notifications and collect relevant time and time zone information.
- Vibra service: methods for triggering and controlling vibrations.
- Device orientation: respond to changes in orientation and discover current orientation.

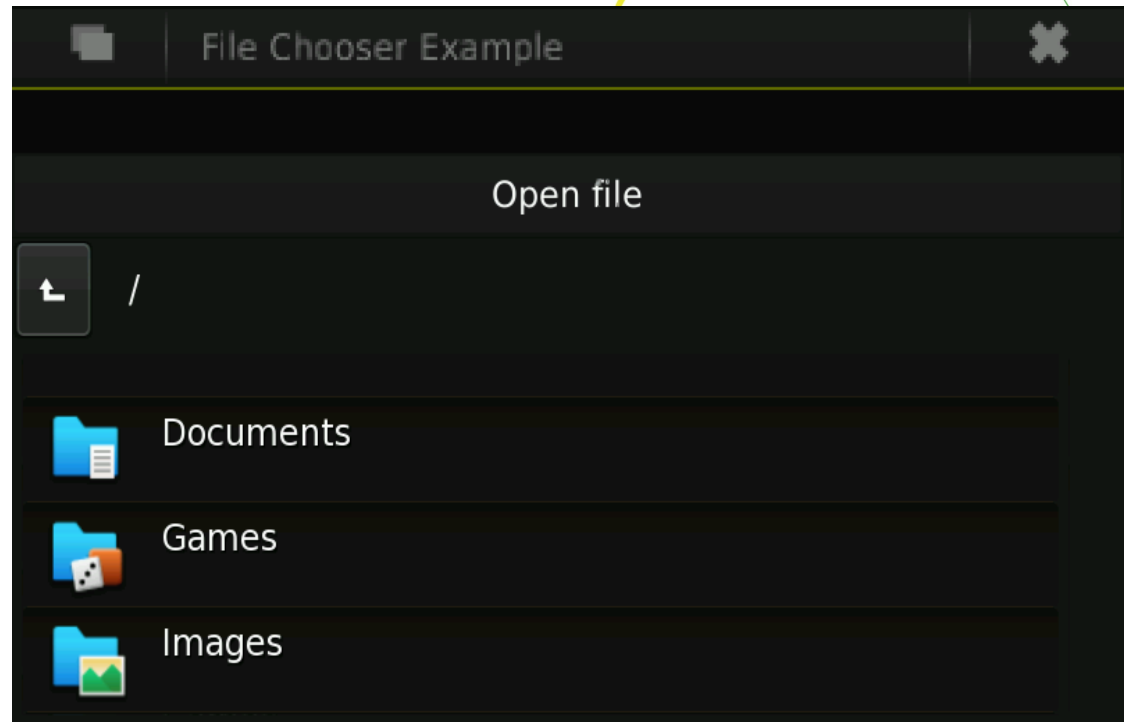
Desktop changes

- Removal of left side Task Navigator and plugins.
 - In home screen there is only application menu button and applets
 - Applications are now in applications menu
 - In application menu all applications are big big thumbable icons
- In next SDK
 - New design for task switching and task handling
 - Renewal of Home and Status bar



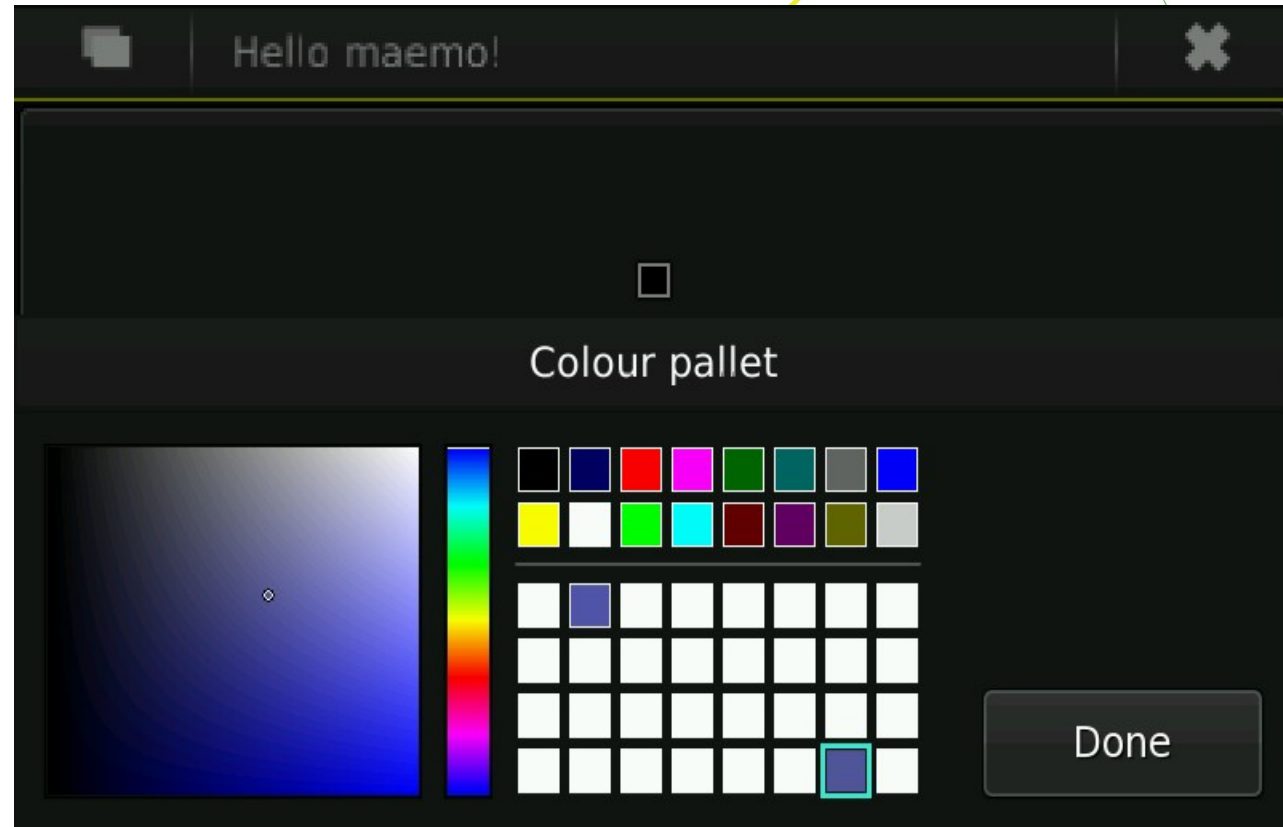
New file chooser dialog

- New thumbable layout
- Kinetic scroll



New dialog style

- Full screen width
- No cancel or close button
close by clicking outside,
screen up to dialog



Brief overview of the Desktop domain



- The Desktop domain comprises of four processes:
- hildon-desktop,
- hildon-home,
- Hildon-status-menu
- Hildon input method server.

The old way hildon desktop

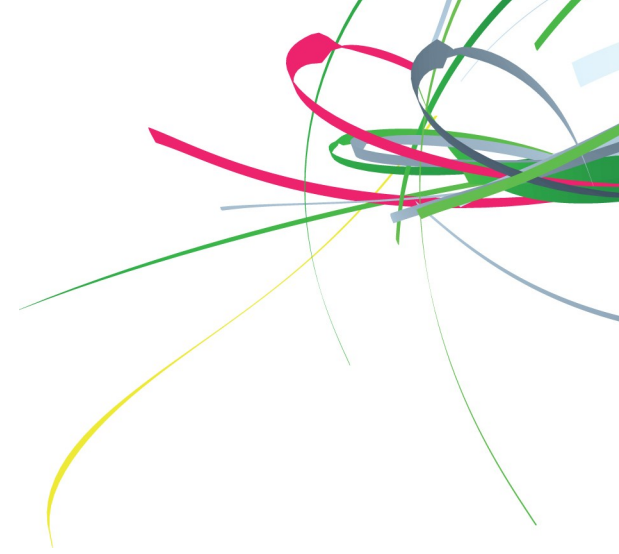
- In Diablo, the single hildon-desktop process implemented the Home for desktop applets, Statusbar for the status applets on top of the screen, Task Navigator for launching and switching between applications, Dbus service for notifications and compositing functionality for transparency in applets.
- Programming errors in these different types of applets often caused hildon-desktop process to crash and restart. After a restart following such a crash, hildon-desktop would only load a 'safe' set of applets, excluding applets that were not part of the sales package.

The Fremantle way

- In Fremantle, the widgets (formerly known as applets) are removed from the process implementing window management, application switching, application launching and drawing of the Home.
- This separation will allow for better reliability in aforementioned functions, due to not having the widgets sharing the same memory space, not having the widgets starting separate threads of execution, and not having the widgets partake in the same Glib main loop as the process that draws the Home, implements application switching and launching etc.
- The separation will also make debugging of widgets easier because the widgets loader processes can be really simple and easily understandable by widget developers.

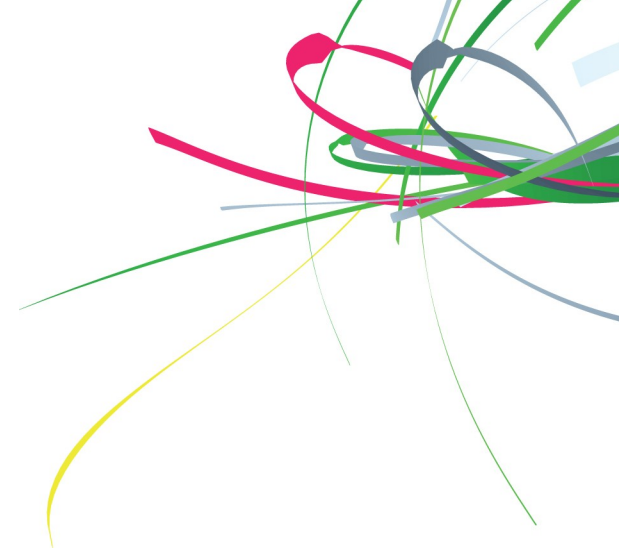
Hildon desktop

- The hildon-desktop process contains a
- compositing window manager
 - There is not much composite effects in alpha SDK
- task launcher (Application Menu)
 - Task launcher is a menu with icons to start applications.
- task switcher (for multitasking).
 - Note: The task switcher is not included in this alpha SDK release.



Hildon Home

- The hildon-home process contains
 - notification manager,
 - the background picture of the desktop
 - graphical widgets,
 - small applications, on top of the background.
 - The widgets are Gtk+ applications implemented as dynamically loadable libraries, and they usually have smaller windows to render their content than 'normal' applications.
 - The widget windows are child windows of hildon-home's top-level window.
- Notification manager provides a systemwide D-Bus service for displaying notifications on the screen.



hildon-status-menu

- The hildon-status-menu process implements the Status Menu and Status Area. Status
- Menu contains Gtk+ applets for displaying miscellaneous information and for controlling system settings.
- Note: This feature is not included in the alpha SDK release.

OpenGL-ES2.0

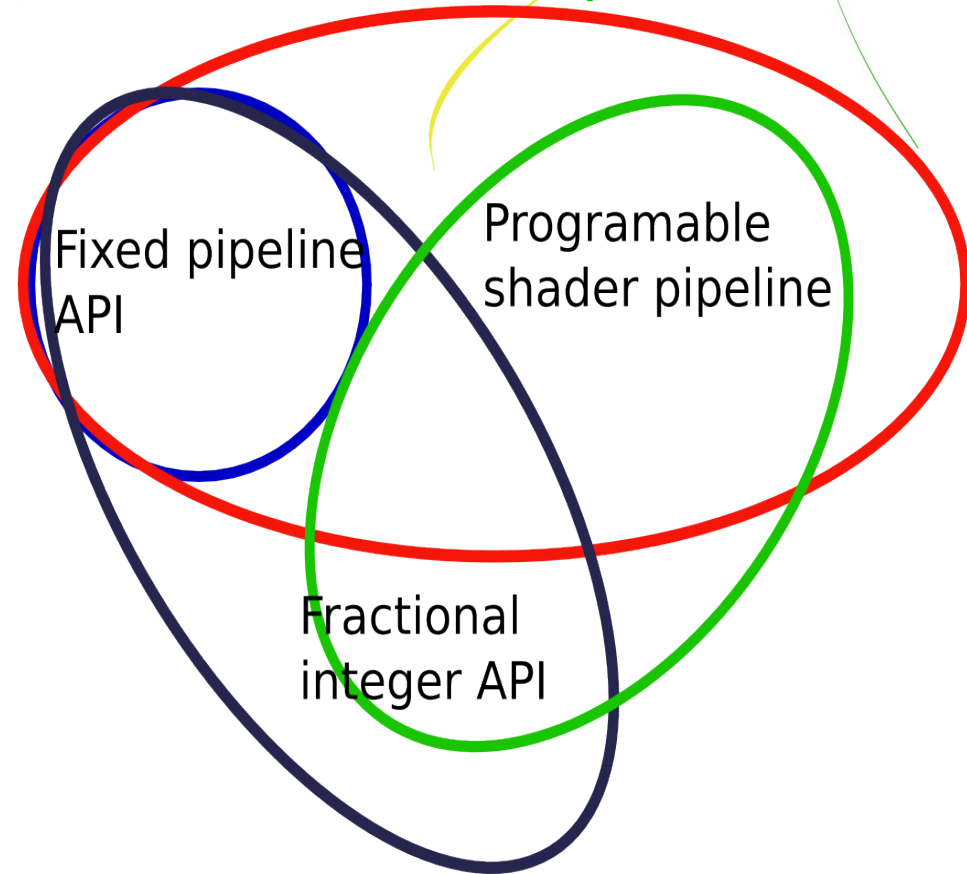
- Hardware accelerated OpenGL-ES2.0 in Omap3 based devices
- Used by
 - Hildon desktop compositing window manager
 - Clutter
 - Qt
- OpenGL 2.0 style programable shaders
-
- Imagination technologies SDK
 - <http://www.imgtec.com/powervr/insider/sdk/KhronosOpenGL-ES2xSGX.asp>
- Books
 - "Mobile 3D graphics with OpenGL-ES and M3G"
 - "OpenGL ES 2.0 programming guide"

OpenGL versions

- OpenGL 1.0 has fixed shaders and fixed function API to using them
- OpenGL 2.0 adds programmable shaders, but fixed function pipeline API is still there for backward compatibility
- OpenGL 3.1 removes support for old fixed function pipeline, so it is very similar to OpenGL ES 2.0
- OpenGL ES 1.0 is based on OpenGL 1.0 with extra redundant API's removed and fixed point (fractional integer) API added
- OpenGL ES 2.0 is based on OpenGL 2.0, so using programmable shaders is mandatory, since all old fixed function API's have been removed

OpenGL 1.0
OpenGL-ES1.0

OpenGL2.0
OpenGL-ES2.0



Qt4.5 for Fremantle

- Qt4.5 Fremantle port released 2.3
 - Home qt4.garage.maemo.org
 - Based on Qt4.5 rc 1
 - Hildon input method (auto completion not implemented)
 - Hildon menus
 - QGTK/Hildon Style (some things needs improvement)
 - OpenGL-ES2.0 support
 - Lot of Fremantle related fixes and workarounds
 - Installable from maemo.org extras-devel repository
- Version based on Qt 4.5 final release soon

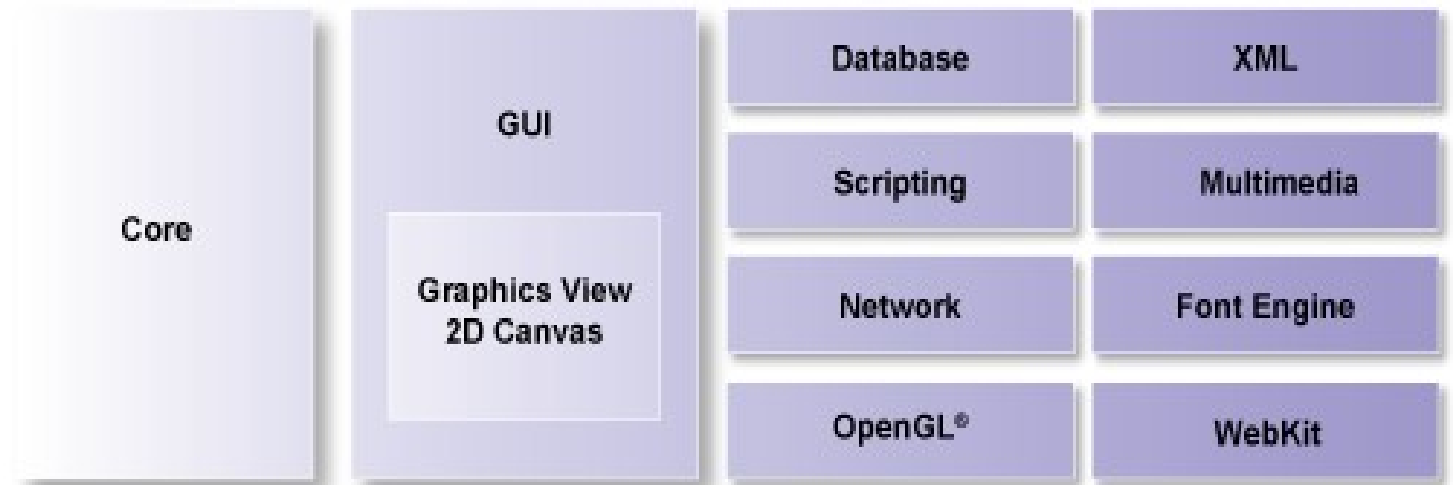


Why Qt in maemo

- Qt is Nokia cross platform toolkit, made by Nokia Qt Software
- Same applications will run with no modifications or just small changes in S60 but also in Desktop Linux Macintosh, Windows and even Windows mobile
- UI needs to be tailored to specific device and device characteristics
- In desktop, big screen and mouse you can have small UI elements and lot of toolboxes visible
- In mobile device users prefer finger UI and large UI elements
- Most of S60 devices has keyboard only

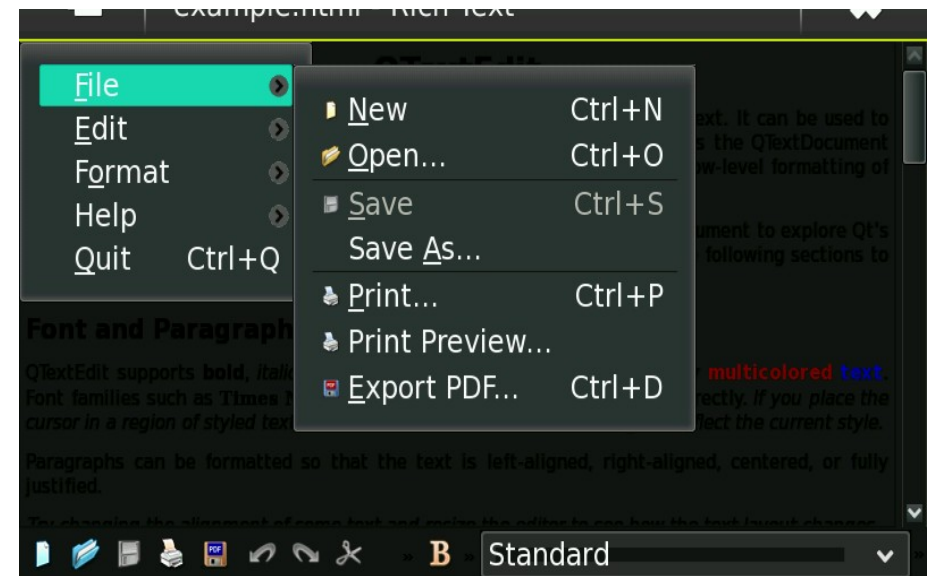
Qt

- Qt is just not a GUI library but complete collection of cross platform libraries
- Integrates features used to be many separate libraries with incompatible API's to common set
-



Hildon menu

- Application main menu is not in application owned window, it is in window manager/desktop owned area
- Application receives X11 Grab transfer event to show menu
- In full screen mode, the menu bar is not visible
- Tablet has hardware menu button, keycode as F4
- No changes are required to existing applications



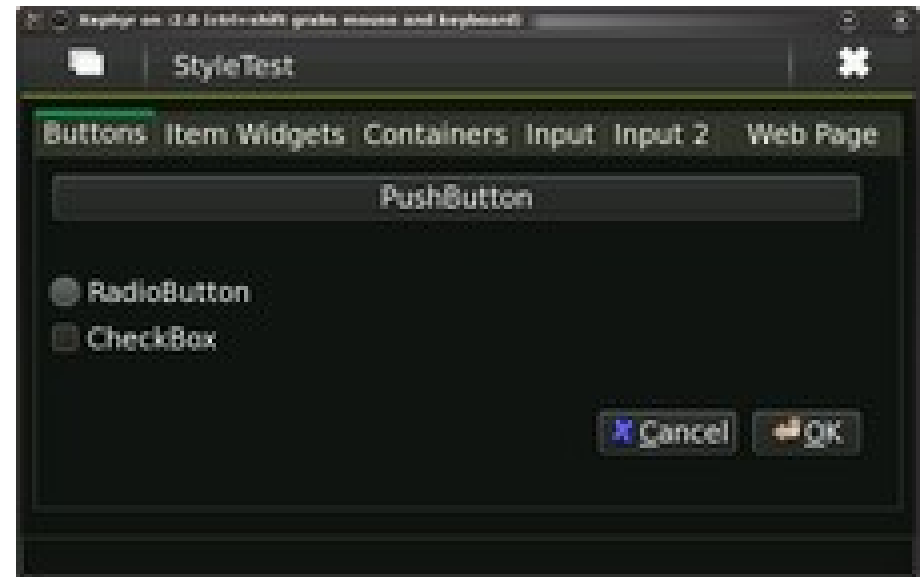
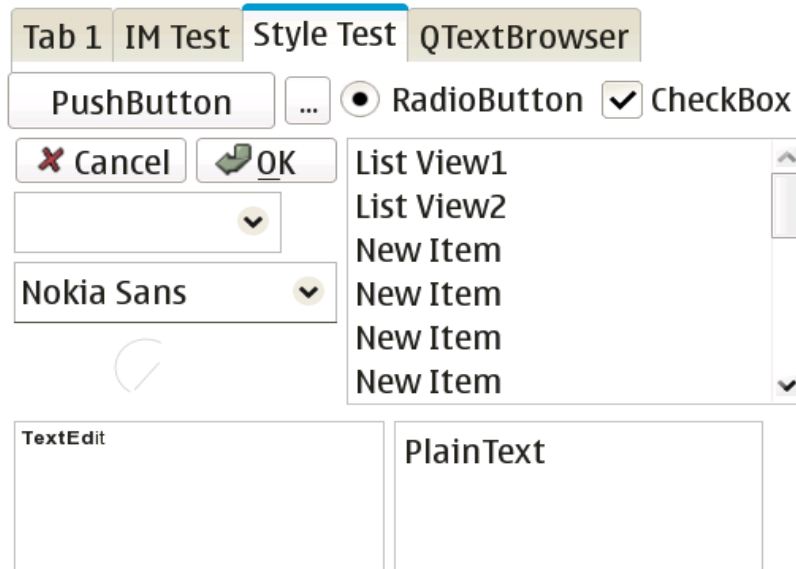
Hildon Input method

- tablet keyboard uses normal X key events
 - there are many mandatory features that need to be supported
- small keyboard has a different modifier configuration
- Modifiers have three modes, “**traditional**” pressed same time with the key, “**sticky**”, pressed before key or “**lock**” pressed twice and modifier is active until pressed third time.
- Fn modifier to access numbers and special characters
- Char modifier to access some characters not in keyboard
- Auto completion mode
- Input mode can be changed by application



Maemo styles

- Maemo uses optimized sapwood theming engine
- Maemo Qt uses sapwood themes
- Themes are optimized for small screen, larger sizes, larger fonts



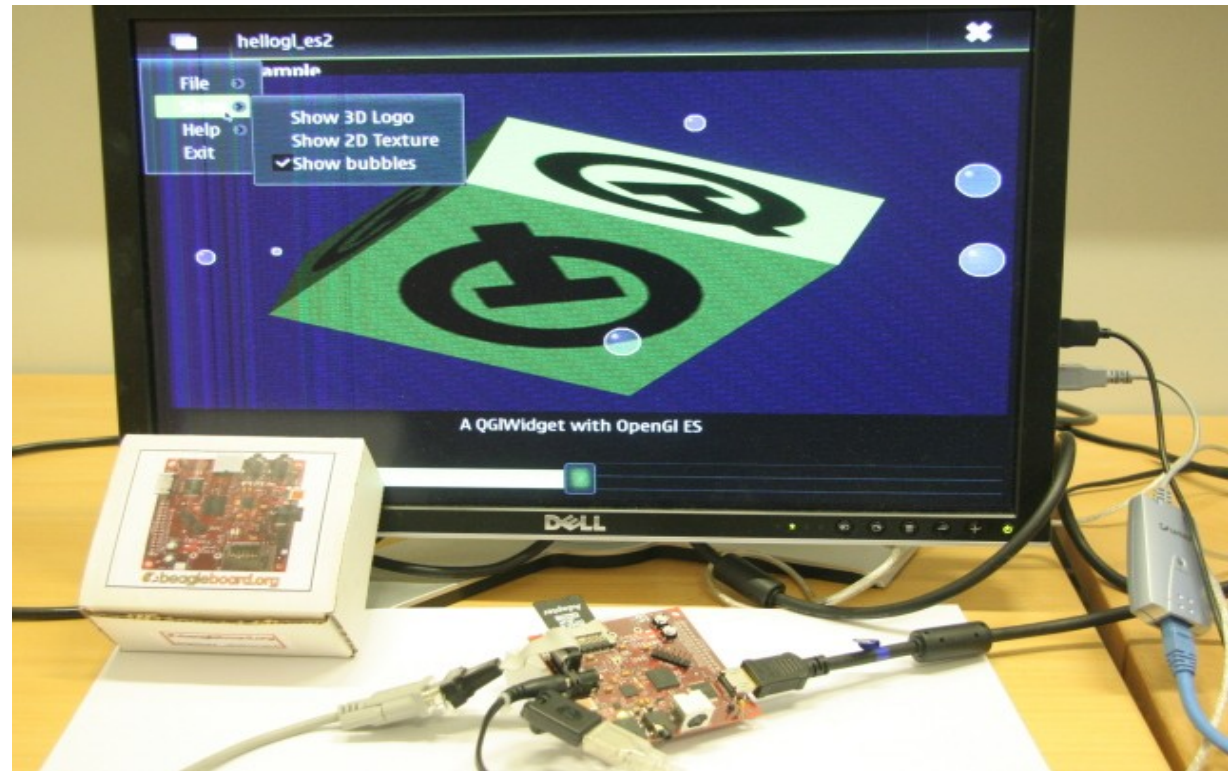
Qt GraphicsView

- Qt way to implement animated UI
- Similar functionality than Clutter
- GraphicsView is a low level toolkit which does not contain any high level widgets people are used to. You can make QGraphicsItems to behave like animated buttons for example or you can embed 2D QT Widgets but they lack lack animated behavior
- You have to manage the layout manually with coordinates, it is not automated



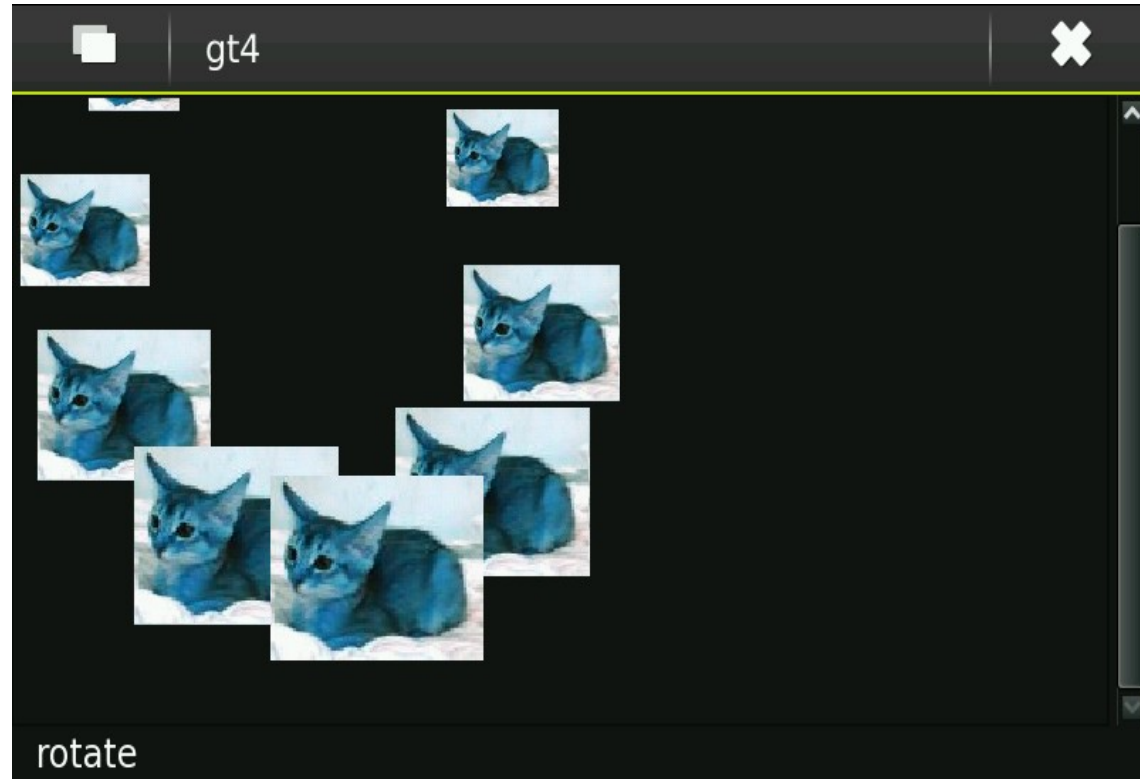
Qt OpenGL-ES 2.0

- As new feature, Qt 4.5 has native OpenGL-ES2.0 support
- GL render for QGraphicsView
 - Transparent to applications
 - Speeds up rendering
- QGLWidget
 - Exposes all features of OpenGL to application
- There are several compatibility issues with OpenGL-ES2.0 driver that needed to fix in maemo port



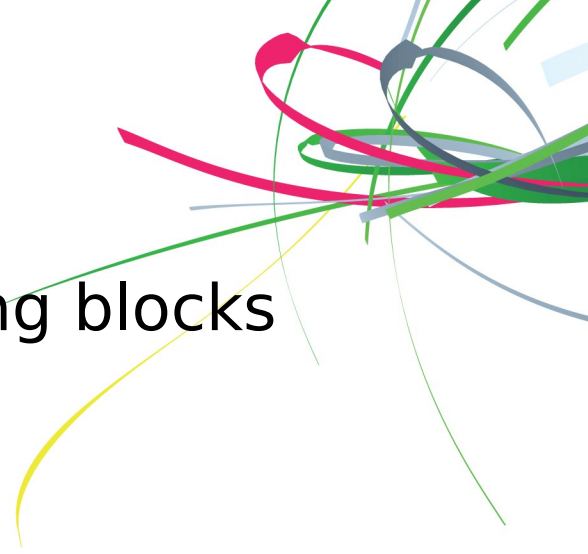
GraphicsView for animatd UI

- QGraphicsScene, QGraphicsItem and QtimeLine
- Supports both OpenGL for desktop and OpenGL-ES in mobile
- GraphicsItems are objects that bahave in stage
 - Move, rotate, scale, change opacity
 - Textures, SVG, vectors
- Timelines and events and events control behavior
- Normal 2D Qt widgets can be embedded and transformed as GraphicsItem



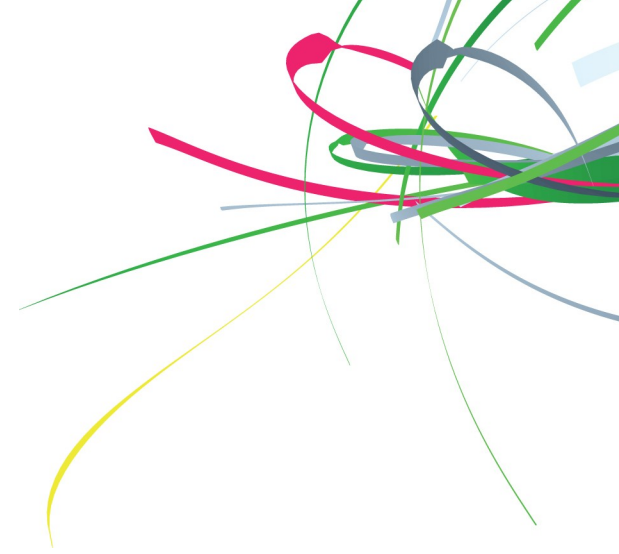
Clutter

- OpenGL based animated GUI toolkit
- Very simple, low level, but still easy. Building blocks which are straightforward to use.
- C / Glib based
- API and conventions resemble that of Gtk+
- Bindings for Python, C++, Vala
- Mobile optimized as design principle
- Supports both OpenGL for desktop and OpenGL-ES in mobile



libcityinfo

- You can get informations such as:
 - city name
 - country name (in which the city is placed)
 - country code
 - time zone
 - geographical position
 - position on the map*
 - locale used in country
- Some functions example
 - `Cityinfo * cityinfo_find_closest (gdouble x, gdouble y)`
 - `gchar * cityinfo_get_name (const Cityinfo *city)`
 - `gchar * cityinfo_get_zone (const Cityinfo *city)`
 - `gchar * cityinfo_get_country (const Cityinfo *city)`
 - `gchar * cityinfo_get_locale (const Cityinfo *city)`



Acceleration sensor

- D-Bus Interface via the Mode Control Entity.
- `#define MCE_DEVICE_ORIENTATION_GET "get_device_orientation"`
- Query the device orientation information.
- Returns:
 - `gchar * portrait/landscape orientation (see mce/mode-names.h for valid portrait/landscape states)`
 - `gchar * on/off stand (see mce/mode-names.h for valid stand states)`
 - `gchar * face up/face down (see mce/mode-names.h for valid facing states)`
 - `dbus_int32_t x axis (unit mG)`
 - `dbus_int32_t y axis (unit mG)`
 - `dbus_int32_t z axis (unit mG)`
-
- `#define MCE_DEVICE_ORIENTATION_SIG "sig_device_orientation_ind"`
- Notify everyone that the device orientation has changed.
- Returns:
 - `gchar * portrait/landscape orientation (see mce/mode-names.h for valid portrait/landscape states)`
 - `gchar * on/off stand (see mce/mode-names.h for valid stand states)`
 - `gchar * face up/face down (see mce/mode-names.h for valid facing states)`

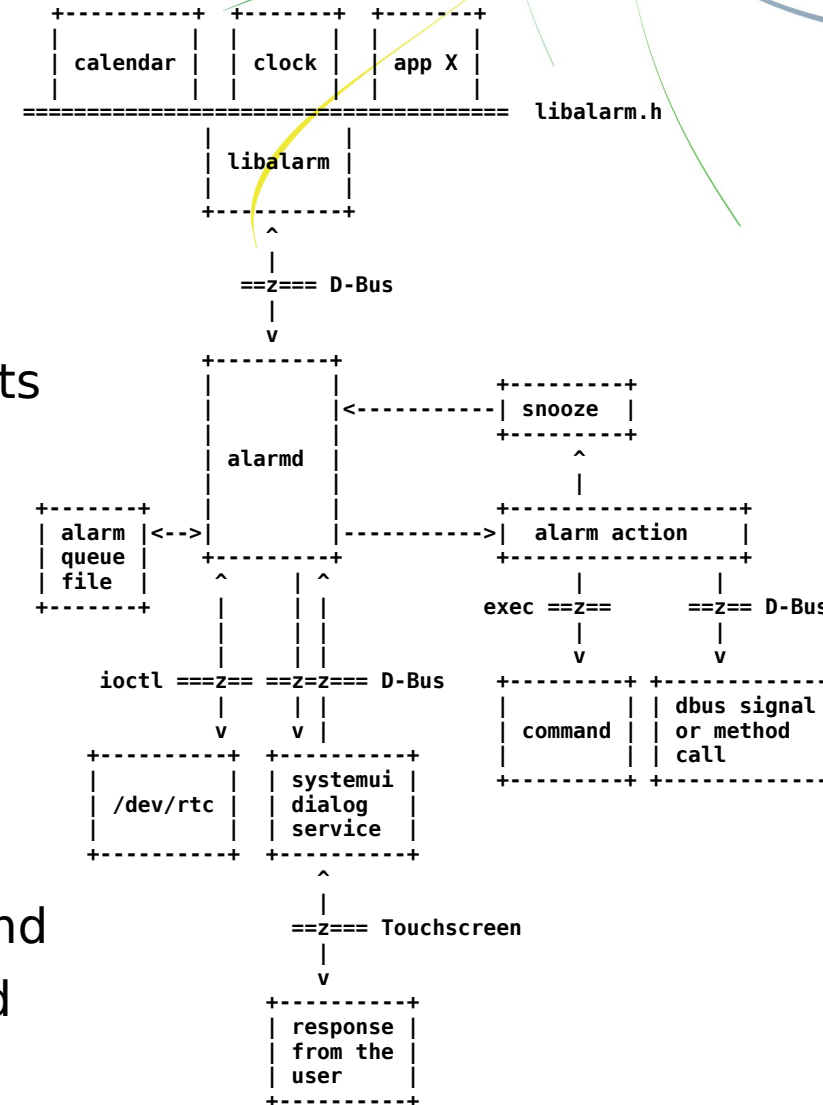
Alarm service

- Alarm service is a subsystem that consists of two parts:

- alarmd - daemon process that keeps track of current alarms
- libalarm - client library for accessing the alarm data

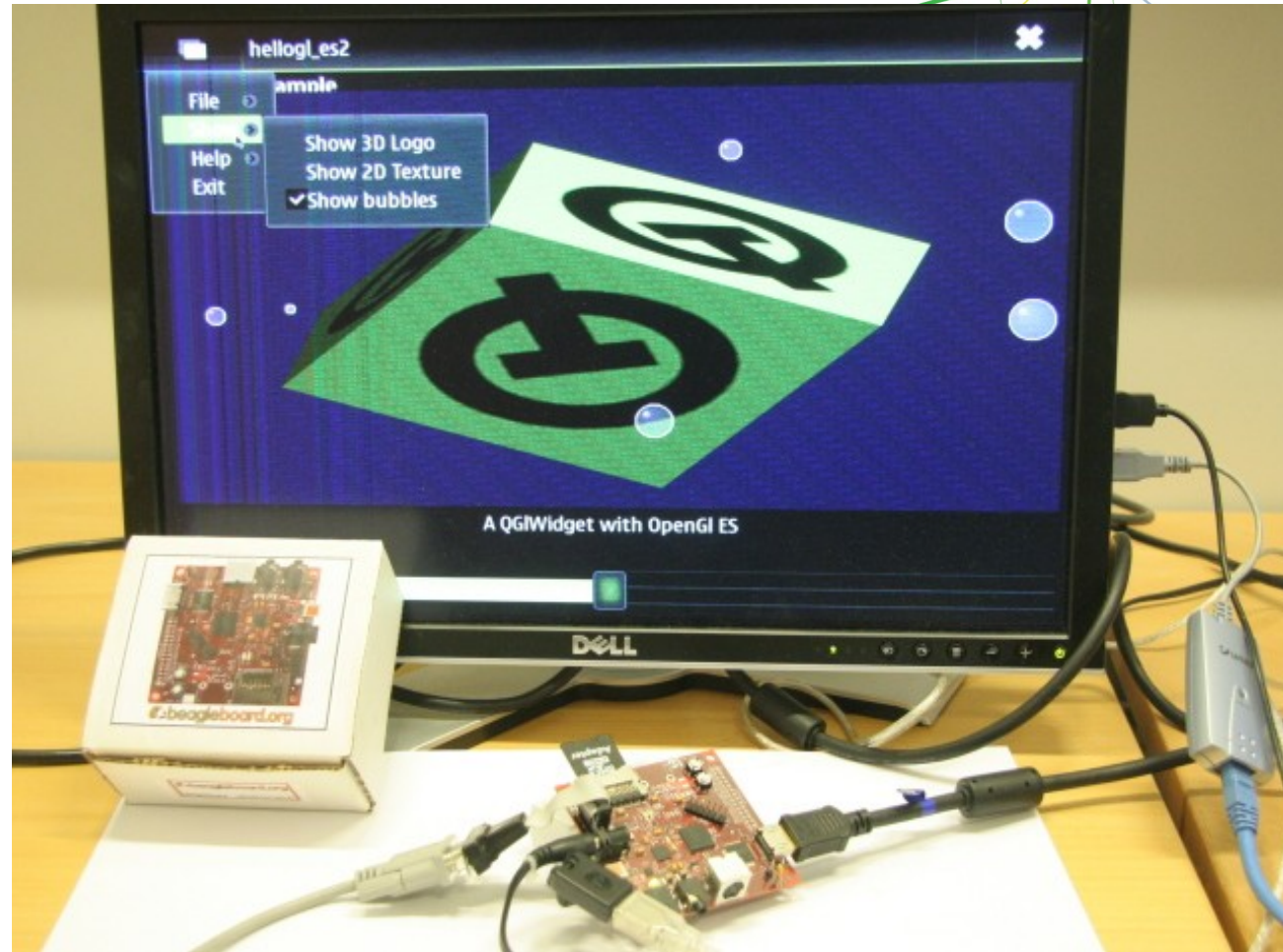
- Architecture

- Alarm daemon supports any number of clients
- Libalarm utilizes D-Bus to communicate with the server
- Alarm daemon accepts request on dbus
- Alarm data is stored to a file
- HW RTC can be used to boot up the device
- Triggered alarms are send to system ui dialog service
- System ui sends user response back to alarmd
- Snooze / Execute / D-Bus action is performed by alarm daemon



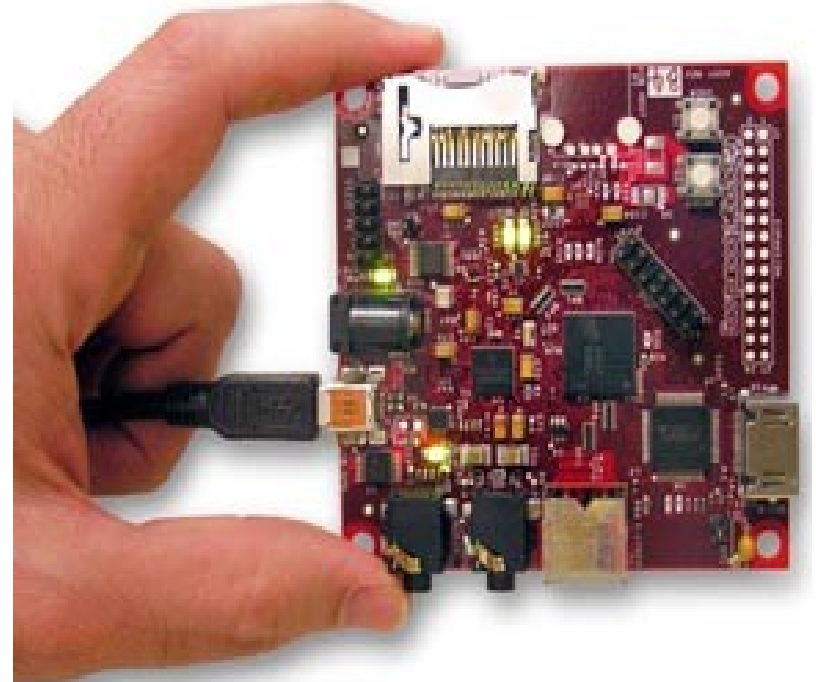
Maemo for Beagleboard

- Beagleboard
- Powered USB HUB
- +5V power supply
- USB keyboard, mouse and Ethernet
- Beagleboard serial cable
- DVI monitor and HDMI->DVI cable
- 2..8Gbyte SD card
- Maemo Fremantle SD card filesystem image



Beagleboard

- www.beagleboard.org
- Omap 3520
- \$149
- Runs our Maemo Fremantle image
- Processor+memory
- USB OTG for peripherals
- HDMI for display
- SD for mass memory
- Serial console



What you don't have in Beagleboard

- Cellular engine
- Wireless connectivity Bluetooth or WiFi
- GPS
- Camera
- Acceleration sensor
- Touchscreen - you can use mouse instead

Where to get maemo beagleboard

An abstract graphic in the top right corner consisting of several overlapping, curved lines in red, green, blue, and yellow, creating a dynamic, swirling pattern.

- Get Beagleboard \$149
- <http://www.beagleboard.org>
- Get Fremantle SDK
- Get Beagleboard files and instructions from:
- <http://maemo-beagle.garage.maemo.org/>
- Maemo beagleboard is community project, you can participate and improve it !!

How to develop ?

- When developer needs to use Fremantle alpha SDK with beagleboard
 - If you are developing application using and needing accelerated graphics
 - Using OpenGL-ES2.0 directly
 - Clutter
 - Qt 4.5 with OpenGL-ES2.0 GraphicsView and QGLWidget
- When develop with with Diablo
 - Using normal GTK/Hildon or Qt